In the Specification

At page 5, lines 10 – 19, please replace the paragraph as follows (underlined denotes replacements additions and strikethough notes deletions):

Currently, the controllers take ownership of a portion of their partner's cache (mirrored cache), which is managed without any coordination with the partner. However, the partner controller does not have any information about the contents of this mirrored cache area until a failover takes place. During failover, information directed to the failed controller is rerouted, or failed over, to a partner controller, wherein the partner controller now becomes aware of the contents of the mirrored cache area. In addition, the coherency model that is currently implemented is based upon coherency at the disks instead of coherency in the caches. Since the disk access time is much slower than the access time for the cache memory, it means that the coherency technique is slower than it would be if it all took place in memory.

At page 9, lines 7 – 13, please replace the paragraph as follows (underlined denotes replacements additions and strikethough notes deletions):

The present invention maintains cache coherency in a storage ~~systems~~system in a manner to minimize the performance degradation to a host system, and to allow the caches to be coherent without requiring data to be written (flushed) to the backing disks. Further, the present invention reduces the amount of messaging necessary to maintain cache coherency. The data storage system is merely one embodiment of the invention and the present invention is applicable to any storage system having or needing cache coherency.

At page 10, lines 7 – 11, please replace the paragraph as follows (underlined denotes replacements additions and strikethough notes deletions):

The set of array controllers 110-118 ~~provide~~provides all the functions needed for parallel operation. This parallel architecture offers several benefits such as: fault tolerance (no central controller to break), performance scalability (no central bottleneck), smooth incremental growth (by simply adding another node), and flexibility.

At page 13, lines 1 – 9, please replace the paragraph as follows (underlined denotes replacements additions and strikethough notes deletions):

Each controller 410, 420 still manages an area in memory 450, 455, respectively, on its partner controller 420, 410, respectively, but the area, which is managed, is dynamic and is done with information about the partner controller 420, 410, respectively. Each controller 410, 420 still manages a free list of mirror locations in a memory location of cache lines (not shown) into which it can copy write data into the partner controller 420, 410, respectively. As will be discussed further below, after the data is copied into a cache line, the partner is informed of the new data so that the cache line may be added to the partners hash table 495, 490, respectively.

At page 13, lines 15 – 19, please replace the paragraph as follows (underlined denotes replacements additions and strikethough notes deletions):

After the completed transfer of the host write 440, the second controller determines which mirror cache line 460 on the first controller 410 to copy the data (second host write) into. Once a determination is made, the data is mirrored [[465]]468 from the second controller's cache line 470 to a mirror cache line 460 on the first controller.

At page 15, lines 10 – 19, please replace the paragraph as follows (underlined denotes replacements additions and strikethough notes deletions):

After the first controller 610 gains ownership of the cache line 680, and associated mirrored cache line 670, the third host write 630 data is moved into what will now be called the cache line 670. Since the cache line 670 on the first controller 610 was the mirror of a cache line 680 on the second controller 620, the cache line 680 on the second controller can be reused as a mirror 690 of the third host write 630. Thus, there is no special type of lookup performed. The third host write 630 data is mirrored to the cache line 680 on the second controller. A message 697, which includes any updates to the cache meta data, ~~is send message~~ is sent from the first controller 610 to the second controller 620 to update at least the dirty bit map in the second controller 620.

At page 16, lines 3 – 11, please replace the paragraph as follows (underlined denotes replacements additions and strikethough notes deletions):

Three additional features may be added to the caching code. The first feature is that since the ownership of the mirror lines 670, 675 can switch between the two controllers 610, 620, a method needs to be put into place to allow the ownership of a mirror line to [[be]] pass to the partner controller 620, 610, respectively. This occurs if one of the controllers 610, 620 starts to take [[to]]too large a percentage of its partner's caches 645, 640, respectively, as mirror cache lines. When this takes place, the controller 610, 620, which owns a large percentage of the cache lines, will need to give them back to its partner controller 620, 610, respectively. Additional features are discussed below.

At page 17, lines 3 – 11, please replace the paragraph as follows (underlined denotes replacements additions and strikethough notes deletions):

Another feature may be to provide a message that indicates when a cache line is cleared, the mirrored information is invalid. For example, when a cache line of the survivor controller 710 is cleared, and that cache line can [[to]] be reused, the mirrored information, for example the mirrored cache line 750 on the replacement controller 730, will have invalid data. Thus, this message is needed when a cache line 760 associated with a mirror cache line 750 has been flushed and is being used for a new (and different LUN, LBA) write operation. The controller 730 of the mirror cache line 750 needs to be informed that the data it currently holds is being replaced and will be temporarily invalid.

At page 17, lines 17 – 22, please replace the paragraph as follows (underlined denotes replacements additions and strikethough notes deletions):

Further, an alternative to the ownership management scheme is to allow any cache line that has had its data flushed to disk to be reused by either of the two controllers 710, 730. Thus, [[a]]the next controller needing a free cache line can claim ownership and reuse the "clean" cache line. In this way the controller that is busiest will be able to hold on to a larger portion of its partner's cache lines and the load on the respective controllers will determine the allocation of cache lines.

At page 18, lines 9 – 22, please replace the paragraph as follows (underlined denotes replacements additions and strikethough notes deletions):

The controller 830 may operate under the control of an operating system. The controller 830 executes one or more computer programs under the control of the operating system. Generally, the operating system and the detector programs may be tangibly embodied in a computer-readable medium or carrier, e.g. one or more of the fixed or removable data storage devices 820, or other data storage or data communications devices. A quantizer 870 may be used for conversion of the analog signals to digital signals between the offset cancelled integrator 830and any connecting digital devices. Both operating system and the computer programs may be loaded from the data storage devices into the memory 840 of the controller 830 for execution by the processor 810. The memory 840 is optional, or may be a memory device embedded or otherwise couple to the controller 830. Both the operating system and the controller programs comprise instructions which, when read and executed by the processor 810, cause the detector to perform the steps necessary to execute the steps or elements of the present invention.